

Sistem Otomatis Pendeteksi Wajah Bermasker Menggunakan *Deep Learning*

Mufid Naufal Baay, Astria Nur Irfansyah, dan Muhammad Attamimi
Departemen Teknik Elektro, Institut Teknologi Sepuluh Nopember (ITS)
e-mail: irfansyah@ee.its.ac.id; attamimi@ee.its.ac.id.

Abstrak—COVID-19 merupakan virus yang telah dinyatakan sebagai pandemi oleh WHO, dan di Indonesia sendiri menetapkan COVID-19 sebagai bencana nasional melalui Keputusan Presiden Nomor 12 Tahun 2020. Sumber utama transmisi dari virus ini berasal dari percikan pernapasan atau droplet yang salah satu pencegahan penyebarannya adalah dengan penggunaan masker. Saat ini, pemerintah sedang memberlakukan new normal. Walaupun beraktivitas di lingkungan luar, protokol kesehatan wajib diikuti dan seluruh masyarakat harus disiplin dalam menjalaninya. Pada studi ini dirancang sebuah sistem otomatis pendeteksi wajah bermasker menggunakan deep learning dalam menjalankan fungsinya. Sistem yang dirancang menggabungkan model *deep learning*, detektor wajah, dan program *tracking* dan *counting* menjadi sebuah sistem otomatis yang dibantu oleh *Graphic User Interface (GUI)* serta sebuah perangkat alarm dan *platform Internet of Things* dalam pemakaiannya. Berdasarkan hasil pengujian yang dilakukan mengikuti batasan masalah yang telah dirumuskan, model memiliki tingkat akurasi klasifikasi pada *dataset test* sebesar 99%. Implementasi pada Raspberry Pi 4 menunjukkan sistem berbasis model deep learning yang telah dibuat sukses melakukan deteksi, *tracking* dan *counting* yang datanya dikirimkan kepada alarm yang dirancang dan sebuah *platform IoT*, Ubidots. Performa deteksi maksimal dicapai saat objek deteksi bergerak 0,7 m/s, pencahayaan ≥ 100 lux, dan penggunaan modul *TensorFlow Lite* pada sistem dengan akurasi sebesar 85,7%. Hasil perbandingan dengan metode deteksi lain menunjukkan karakterisasi model *deep learning* memiliki akurasi deteksi sebesar 82%, lebih tinggi dari metode *Haar Classifier* dengan akurasi 53%.

Kata Kunci—Covid-19, *Computer Visions*, *Face Detection*, *Deep Learning*, *IoT*.

I. PENDAHULUAN

CORONA Disease 2019 atau yang lebih awam dikenal COVID-19, telah dinyatakan *World Health Organization (WHO)* sebagai sebuah pandemi. Di Indonesia, penyebaran COVID-19 sudah semakin meluas yang diiringi dengan bertambahnya jumlah kasus positif serta jumlah kematian. Situasi ini mempengaruhi seluruh aspek kehidupan masyarakat baik itu ekonomi, sosial, budaya, pertahanan, keamanan, dan politik. Oleh karena itu, Presiden menetapkan melalui Keputusan Presiden Nomor 12 Tahun 2020 tentang Penetapan Bencana Nonalam Penyebaran COVID-19 sebagai bencana nasional.

Sumber utama transmisi dari COVID-19 berasal dari percikan pernapasan atau *droplet* yang dikeluarkan saat manusia berbicara, batuk, atau bersin. Ukuran droplet umumnya sebesar 5um sampai 10um [1]. Salah satu bentuk pencegahan penyebaran dan perlindungan terhadap penyebaran adalah dengan pemakaian masker. Pada sebuah studi sehingga mengurangi penyebaran jika upaya

pencegahan tersebut dilakukan dengan benar. Tetapi hal tersebut hanya berlaku apabila sebagian besar sampai seluruh masyarakat mengikutinya, karena apabila banyak masyarakat yang tidak mengikutinya, pemakaian masker untuk pencegahan menjadi tidak terlalu efektif [2].

Pemerintah memberlakukan *New Normal*, yaitu upaya menjaga produktivitas masyarakat di tengah pandemi COVID-19 Setelah usainya pelaksanaan Pembatasan Sosial Berskala Besar (PSBB) di Indonesia agar roda ekonomi tetap berjalan sampai ditemukannya vaksin dengan standar internasional untuk pengobatan COVID-19. Pada new normal ini masyarakat harus mengubah pola hidup menjadi lebih sehat dan tetap memberlakukan protokol kesehatan seperti menjaga jarak dan selalu memakai masker pada lingkungan umum untuk mencegah penyebaran COVID-19 selagi beraktivitas di luar. Oleh karena itu dibutuhkan kedisiplinan dari setiap masyarakat untuk mengikuti protokol yang ada.

Tujuan dari studi ini adalah merancang sebuah sistem otomatis pendeteksi wajah bermasker dan tidak bermasker sebagai salah satu bentuk upaya meningkatkan kesadaran dan kedisiplinan masyarakat dalam menjalani kehidupan new normal. Sistem akan mendeteksi dan mengklasifikasi wajah apakah seseorang tersebut memakai masker atau tidak pada tempat yang dipantau dan memberikan informasi berupa jumlah dari masing-masing hasil klasifikasi. Dengan adanya sistem ini diharapkan dapat meningkatkan kedisiplinan masyarakat dalam melaksanakan new normal di Indonesia.

II. TINJAUAN PUSTAKA

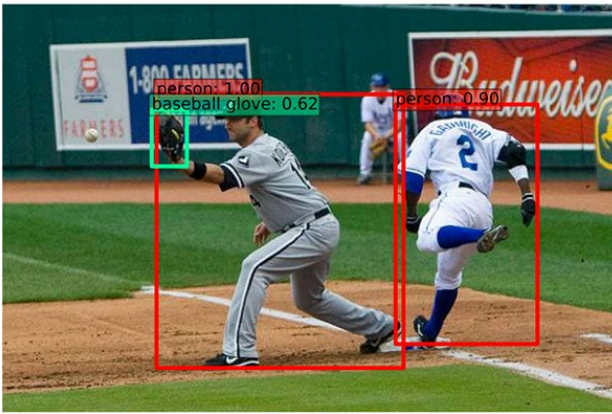
A. *MobileNetV2*

MobileNets merupakan salah satu arsitektur *convolutional neural network (CNN)* yang cocok digunakan untuk mengatasi masalah keterbatasan daya komputasi pada sebuah sistem dalam menjalankan tugas seperti deteksi objek menggunakan *deep learning* dengan penggunaan layer konvolusi yang dibagi menjadi *depthwise convolution* dan *pointwise convolution* [3].

Depthwise separable convolution merupakan fitur yang dihadirkan oleh model *MobileNetV1*. Ini merupakan sebuah blok pada deep learning yang dari dari *depthwise convolution* dan *pointwise convolution*. Tujuan dari *depthwise separable convolution* adalah mereduksi kebutuhan daya komputasi dan ukuran dari model yang dibuat

Kebutuhan daya komputasi yang dibutuhkan oleh arsitektur *MobileNet* adalah:

$$D_K \cdot D_K \cdot M \cdot D_F + M \cdot N \cdot D_F D_F D_F \quad (1)$$



Gambar 1. Contoh Deteksi Menggunakan Metode Single Shot Detector.



Gambar 2. Keseluruhan Hardware yang Dirancang.

D_K Adalah ukuran atau dimensi dari kernel. M adalah jumlah kanal input, N adalah jumlah dari kanal output dan D_F adalah ukuran dari fitur atau filter. Sedangkan untuk konvolusi standar tanpa arsitektur MobileNet memiliki kebutuhan daya komputasi sebagai berikut:

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F D_F \quad (2)$$

Operasi konvolusi dilakukan oleh setiap filter dengan seluruh kanal pada konvolusi standar. Dengan menerapkan faktorisasi oleh *depthwise separable convolution* yang dilakukan oleh MobileNet, kebutuhan daya komputasi yang dibutuhkan akan dapat direduksi

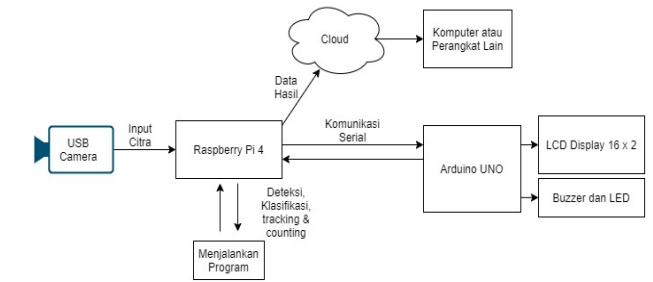
$$\frac{D_K \cdot D_K \cdot M \cdot D_F + M \cdot N \cdot D_F D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F D_F} = \frac{1}{N} + \frac{1}{D_K^2} \quad (3)$$

Rumus menjelaskan apabila ukuran kernel $D_K \times D_K$ adalah 3×3 maka kebutuhan komputasinya akan tereduksi kurang lebih 9 kali lebih sedikit.

MobileNetV2 merupakan versi kedua dari MobileNets, masih menggunakan *depthwise* dan *pointwise convolution*. Pada versi ini ada dua fitur tambahan yaitu *linear bottleneck* serta *shortcut connections* antar *bottlenecks* untuk menghilangkan aktivasi ReLU pada akhir proses inverted residual block sehingga informasi yang diberikan tidak ada yang hilang [4].

B. Transfer Learning

Metode *transfer learning* yang digunakan pada penelitian ini adalah metode *fine-tuning*. *Fine-tuning* adalah sebuah proses yang memiliki beberapa tahap:



Gambar 3. Diagram Blok Sistem.

Tabel 1. Dataset Pertama [8]

Label	Jumlah Gambar	Total Gambar Dataset	Format Gambar
With_Mask	690	1376	JPG
Without_Mask	686		

Tabel 2. Dataset Kedua [9]

Label	Jumlah Gambar	Total Gambar Dataset	Format Gambar
With_mask	680	853	JPG
Without_mask	148		
Wear_mask_incorrectly	25		

1. Menghilangkan FC layer pada bagian akhir network dimana tempat prediksi kelas label dibuat.
2. Membuat FC layer yang baru dan menginisialisasinya pada tempat FC yang lama.
3. Melakukan *'freezing'* pada layer konvolusi agar weight-weight yang dilakukan untuk ekstraksi fitur pada layer itu tidak berubah.
4. Melakukan proses *training* hanya pada FC layer.
5. Perlahan melakukan *'unfreezing'* pada sebagian atau semua layer konvolusi diiringi dengan *training* kedua menggunakan *learning rate* yang sangat kecil.

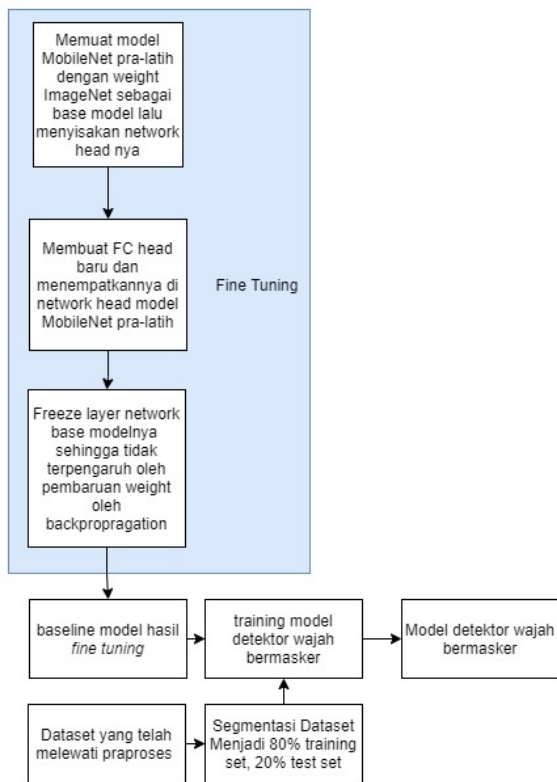
Fine-tuning merupakan metode yang sangat cocok untuk diterapkan untuk membuat klasifikasi citra menggunakan dataset yang dikumpulkan sendiri dengan memanfaatkan model ML pra-trlatih serta dapat membantu membuat model tanpa harus membuatnya dari awal [5].

C. Single Shot Detector

Single Shot Detector (SSD) merupakan sebuah metode deteksi objek pada sebuah citra input menggunakan *single deep neural network*. Metode deteksi objek ini merupakan salah satu algoritma deteksi objek yang populer digunakan karena kemudahan implementasi serta rasio kecepatan komputasi dan akurasi yang dimilikinya. SSD adalah hanya perlu mengamati satu *'shot'* atau satu kali pembacaan *frame* untuk mendeteksi objek-objek pada *frame* tersebut. Pada input citra 300×300 , SSD mencapai 72.1% pada nilai *mean average precision (mAP)* pada VOC2007 dengan 59 FPS. Untuk input citra 500×500 , SSD mencapai 75.1% pada nilai mAP dengan 22 FPS. Keduanya di test pada Nvidia Titan X yang mengalahkan model SOTA Faster R-CNN (73.2% mAP pada 7 FPS) dan YOLOv1 (63.4% mAP pada 45 FPS) [6].

D. Deep Learning

Deep learning merupakan salah satu subset dari *machine learning* atau pembelajaran mesin yang terinspirasi dari



Gambar 4. Tahap Fine-Tuning Model

struktur dari otak manusia. Seperti halnya manusia membuat keputusan dengan menganalisa menggunakan rangkaian pemikiran logis yang terstruktur, *deep learning* mengimitasinya dengan sebuah algoritma yang dinamakan *neural network* [7]. DL memungkinkan mesin untuk belajar dari data. NN yang digunakan pada DL adalah *Deep Neural Network (DNN)* yaitu NN dengan lebih dari satu *hidden layer* diantara input dan output. Penggunaan jumlah *hidden layer* dan *neuron* ditentukan sesuai kebutuhan. DNN pada umumnya adalah *feedforward network* yaitu aliran data dari input ke output tidak terdapat pengulangan.

III. PERANCANGAN SISTEM

A. Diagram Blok Sistem

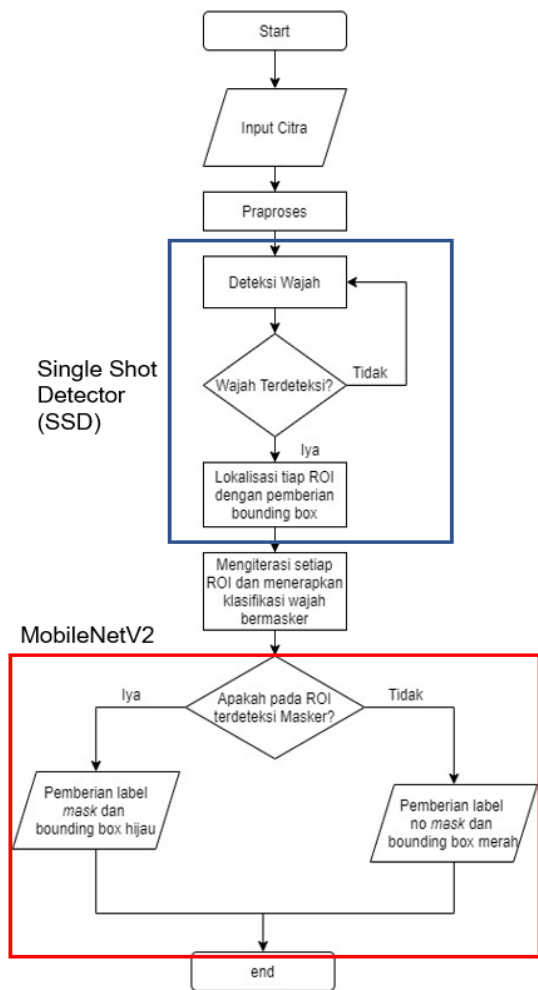
Sistem yang akan dirancang berfungsi mendeteksi otomatis wajah bermasker dengan input berupa video dari kamera lalu akan diproses pada komputer mini raspberry pi 4 dengan tahap deteksi wajah, klasifikasi wajah bermasker, dan tracking yang akan berjalan satu kesatuan dalam sebuah *pipeline*, data hasil klasifikasi lalu dikirimkan melalui internet agar dapat dianalisa pada komputer. Dengan sistem seperti ini memungkinkan analisa yang lebih fleksibel.

B. Dataset

Ada dua *dataset* yang digunakan dalam penelitian ini yang masing-masing berasal dari seorang pengguna github, bernama Prajna Bhandary serta dari situs kaggle.com [8] yaitu situs untuk mencari dan mempublikasi *datasets* untuk para pengembang *machine learning*. Keterangan dari kedua *datasets* akan dijelaskan pada Tabel 1 dan Tabel 2.

C. Praproses Dataset

Dataset yang dipakai harus melalui tahap pra-proses agar citra yang ada didalam *dataset* dapat diproses dengan baik



Gambar 5. Flowchart Deteksi

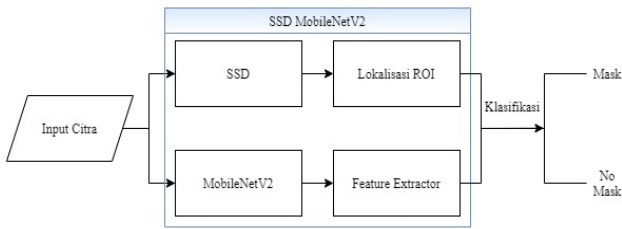
pada tahap selanjutnya. Ada 3 tahap pra-proses dataset yaitu *resizing* menjadi 224 x 224 [9], *grayscale*, dan augmentasi data [10].

D. Pembuatan Model Deep Learning dengan Fine-Tuning

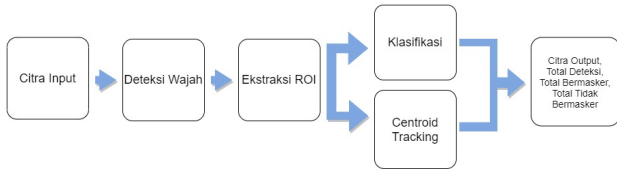
Keseluruhan pembuatan model akan dibuat menggunakan pustaka TensorFlow pada Bahasa pemrograman python. Proses ini dimulai dengan memuat MobileNet pra-terlatih dengan *weight ImageNet* [11]. MobileNet pra-terlatih ini kemudian dibuang *network head* nya serta menjadikannya model dasar. Selanjutnya penulis mengonstruksi sebuah *fully connected head* yang akan disambungkan dengan model dasar. mengikuti langkah yang telah disebutkan pada subbab C. Model patokan MobileNet serta dataset yang telah disegmentasi digunakan pada proses *training* untuk menghasilkan model detektor wajah bermasker yang memiliki label output *mask* dan *no mask*.

E. Proses Deteksi Menggunakan model Deep Learning

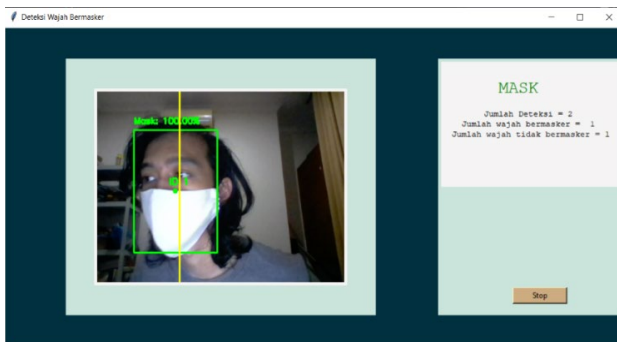
Model klasifikasi yang didapatkan *fine tuning* arsitektur MobileNetV2 tidak berdiri sendiri. Model klasifikasi yang didapatkan dengan melatih arsitektur MobileNetV2 akan digunakan sebagai *feature extractor* untuk mengklasifikasikan wajah bermasker dan tidak bermasker. Setelah itu, model yang didapatkan akan dikombinasikan dengan detektor objek *Single Shot Detector (SSD)* sebagai lokalisasi *Region of Interest (ROI)*. ROI yang didapatkan lalu akan dijadikan sebagai input model klasifikasi wajah bermasker.



Gambar 6. Diagram Fungsi SSD dan MobileNetV2



Gambar 7. Pipeline Fungsi Sistem.



Gambar 8 Graphic User Interface (GUI) Sistem

MobileNetV2 berperan sebagai *feature extractor* pada diagram Gambar 4. ROI yang ditandai dengan pemberian *bounding box* akan dijadikan sebagai input klasifikasi pada model MobileNetV2 yang telah dirancang. Sebuah sistem yang dapat mendeteksi wajah bermasker secara otomatis didapatkan dengan memanfaatkan kombinasi MobileNetV2 dan SSD.

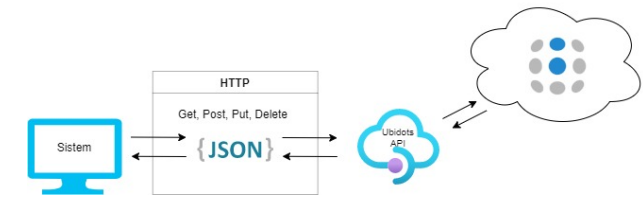
F. Centroid Tracking

Sistem otomatis pendeteksi wajah bermasker yang dirancang diinginkan untuk mempunyai kemampuan untuk menghitung klasifikasi jumlah wajah bermasker yang terdeteksi tanpa pengulangan perhitungan pada wajah yang sama.

Algoritma tracking yang akan digunakan adalah algoritma *centroid tracking*. Algoritma tracking ini memanfaatkan *bounding box* dari detektor wajah SSD pada tahap sebelumnya. Dengan memanfaatkan *bounding box* yang diterapkan pada setiap frame, dapat dihitung titik tengah yang akan dijadikan “centroid” sebagai titik patokan untuk penerapan *object id* unik pada objek yang terdeteksi. Agar tracking dapat terjadi, algoritma ini memanfaatkan jarak euclidean antara *centroid* objek yang sudah terdeteksi dan centroid objek baru yang ada pada frame berikutnya. Dengan menggunakan koordinat (x,y) dari *centroid*, dapat digunakan untuk mencari euclidean distance antar centroid objek pada frame.

$$d(x,y)^2 = (x_1 - y_1)^2 + (x_2 - y_2)^2 \tag{1}$$

Perlu diperhatikan keseluruhan proses tracking akan terus berjalan selama program menerima input video serta framenya dapat terbaca. Setelah objek yang terdeteksi hilang dari frame yang terbaca oleh program dalam jangka waktu



Gambar 9. Pemanfaatan API Ubidots dengan Protokol HTTP.

Tabel 3. Kontrol Aktuator Alarm.

Gambar		
	Display LCD	No Mask Detected Pakai Maskermu!
Data	'0'	'1'
Keterangan	LED Merah dan Buzzer Aktif	LED Biru Aktif

Tabel 4. *Hyperparameter* Evaluasi Model

Hyperparameter	Value
Learning rate	1e ⁻⁴
Epoch	20
Batch Size	32

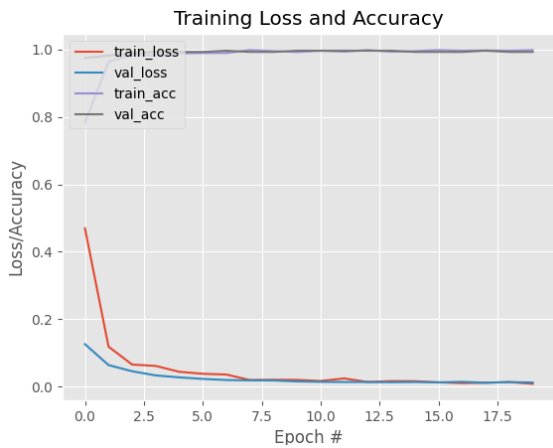
tertentu, algoritma akan otomatis menghapus centroid beserta ID objek dari frame sehingga memungkinkan objek baru terdeteksi.

G. Sistem Otomatis Pendeteksi Wajah Bermasker

Keseluruhan rancangan yang dibuat pada subbab A sampai D akan dijadikan satu untuk mendapatkan sebuah sistem yang dapat mendeteksi wajah bermasker secara otomatis dan menampilkan informasi yang didapat. Untuk mencapai hal itu, penulis membuat sebuah GUI dan pemanfaatan sebuah platform IoT agar sistem dapat dilakukan untuk melakukan monitoring.

Penerapan GUI bertujuan mempermudah sistem pendeteksi wajah bermasker agar dapat digunakan untuk monitoring. Informasi yang ditampilkan berdasarkan deteksi pada frame sebelah kiri. Garis vertikal berwarna kuning akan menjadi tempat patokan dimana deteksi dihitung. Saat centroid dari objek terdeteksi yang diindikasikan dengan ID objek beserta titik berwarna hijau melewati garis kuning, program akan melakukan proses pencacahan. Proses ini akan menerima input klasifikasi hasil prediksi berupa bermasker dan tidak bermasker yang terhubung dengan ID objek unik lalu output akan ditampilkan pada frame sebelah kanan berupa hasil klasifikasi dan total hasil klasifikasi serta total wajah terdeteksi.

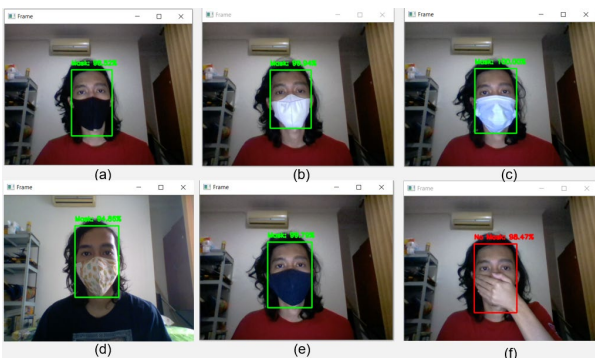
Agar informasi yang diperoleh dari sistem pendeteksi dapat diakses dimana saja secara fleksibel, konsep yang akan diterapkan untuk melakukan hal itu adalah penggunaan *Internet of Things* (IoT). Platform IoT yang akan digunakan untuk studi ini adalah Ubidots. Perancangan sistem ini akan



Gambar 10. Learning Curve Evaluasi Model

Tabel 5. Classification Report dari Evaluasi Model

Label	Precision	Recall	F1-score	Support	Accuracy
mask	0,99	1,00	0,99	138	0,99
no_mask	1,00	0,99	0,99	138	0,99



Gambar 11. Jenis Masker yang diuji (a) warna hitam (b) warna putih (c) masker medis (d) masker bercorak (e) masker biru gelap (f) pengecoh dengan tangan

memanfaatkan fitur Ubidots yang berupa pendukung protokol HTTP, penggunaan API, penyimpanan data, dan visualisasi data. Data berupa total deteksi, total wajah bermasker, total wajah tidak bermasker akan dikirimkan memanfaatkan token API yang disediakan oleh Ubidots untuk setiap pengguna dalam bentuk JSON. Sistem akan melakukan request protocol HTTP melalui pustaka bahasa pemrograman python yaitu *requests* lalu post data JSON kepada platform Ubidots.

H. Perancangan Aktuator Alarm

Aktuator yang digunakan pada studi ini berfungsi untuk menampilkan peringatan berdasarkan hasil deteksi yang diperoleh dari sistem. Aktuator akan dikontrol oleh sebuah mikrokontroler Arduino Uno dalam menjalankan fungsinya. Data yang dikirimkan secara serial akan digunakan untuk kontrol pada mikrokontroler. Raspberry Pi akan mengirimkan data ‘0’ saat masker tidak terdeteksi, data ‘1’ saat masker terdeteksi dimana outputnya akan ditampilkan sesuai pada Tabel 3.

IV. PENGUJIAN DAN ANALISA

A. Evaluasi Model Klasifikasi Deep Learning

Evaluasi model klasifikasi wajah bermasker dilakukan untuk mengetahui tingkat akurasi dari model yang telah

Tabel 6. Pengujian Jenis Masker Metode Deep Learning dan Haar Classifier

Masker	Asli	Prediksi	Status	
			DL*	HC**
A	Mask	-	Sukses	Gagal
B	Mask	Mask	Sukses	Sukses
C	Mask	Mask	Sukses	Sukses
D	Mask	-	Sukses	Gagal
E	Mask	Mask	Sukses	Sukses
F	No Mask	No Mask	Sukses	Sukses

*CR – Deep Learning

**HC – Haar Classifier

Tabel 7. Pengujian Sudut Deteksi Metode Deep Learning dan Haar Classifier

Sudut	Status	
	DL	HC
-45°	Tidak Terdeteksi	Tidak Terdeteksi
-30°	Terdeteksi	Tidak Terdeteksi
0°	Terdeteksi	Terdeteksi
30°	Terdeteksi	Tidak Terdeteksi
45°	Tidak Terdeteksi	Tidak Terdeteksi

Tabel 1

Pengujian Jarak dan Pencahayaan Deteksi Metode Deep Learning dan Haar Classifier

Cahaya	Jarak (cm)	Status	
		DL	HC
Redup ≤ 10 lux	20	Terdeteksi	Tidak Terdeteksi
	40	Terdeteksi	Terdeteksi
	60	Terdeteksi	Terdeteksi
	80	Tidak Terdeteksi	Terdeteksi
	100	Tidak Terdeteksi	Tidak Terdeteksi
Sedang 20 – 50 lux	20	Terdeteksi	Tidak Terdeteksi
	40	Terdeteksi	Terdeteksi
	60	Terdeteksi	Terdeteksi
	80	Tidak Terdeteksi	Terdeteksi
	100	Tidak Terdeteksi	Terdeteksi
Terang ≥ 100 lux	20	Terdeteksi	Tidak Terdeteksi
	40	Terdeteksi	Terdeteksi
	60	Terdeteksi	Terdeteksi
	80	Terdeteksi	Terdeteksi
	100	Tidak Terdeteksi	Terdeteksi

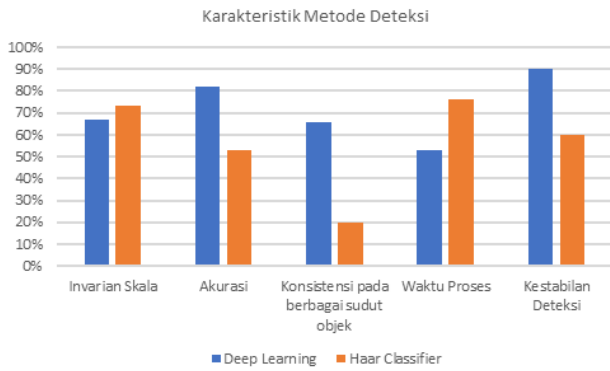
dilatih. Pada pengujian ini dilakukan pengujian dengan membagi dataset menjadi 80% training set dan 20% test set untuk evaluasi model.

Pada subbab ini akan diuji parameter-parameter akurasi seperti *precision*, *recall*, dan *F-1 score*. Selain itu akan dilakukan evaluasi pada setiap epoch pada tahap pelatihan agar didapatkan *training loss* dan *validation loss*. tujuan dari pengujian ini agar didapatkan model yang dapat mengklasifikasi secara general dan menghindari *overfitting*.

Proses pelatihan model menggunakan metode *fine-tuning* untuk melatih model pra-terlatih dengan weight ImageNet. Hyperparameter algoritma yang digunakan pada proses pembelajaran model dijelaskan pada Tabel 4 model adalah *learning rate*, *epoch*, dan *batch size* [12].

B. Karakterisasi Model Deep Learning

Karakterisasi model DL yang dibuat bertujuan untuk mengetahui kemampuan model dalam deteksi dan klasifikasi objek pada berbagai macam kondisi. Kemudian hasil dengan metode DL dibandingkan dengan deteksi menggunakan metode haar classifier [13] untuk mengetahui performa model yang telah dibuat dengan metode yang sudah pernah dipublikasikan dalam menjalankan fungsinya.



Gambar 12. Karakteristik Metode Deteksi *Deep Learning* dan *Haar Classifier*.



Gambar 13. Lingkungan Uji.

Pengujian yang dilakukan adalah pengujian kemampuan metode DL dan HC dalam mendeteksi berbagai macam jenis masker, berbagai sudut deteksi, berbagai jarak deteksi, berbagai iluminasi pada objek, dan berbagai citra uji.

Berdasarkan karakteristik yang telah divisualisasikan pada Gambar 12, model DL yang dirancang mempunyai akurasi deteksi yang lebih tinggi sebesar 82% dari metode haar classifier 53% walaupun waktu proses yang dilakukan sedikit lebih lama 23%. Namun pada sistem yang dirancang oleh Penulis dibutuhkan sebuah metode deteksi yang stabil terhadap berbagai sudut deteksi serta memiliki kestabilan dan akurasi yang tinggi. Oleh karena itu, model DL yang dibuat akan dipilih untuk digunakan pada sistem yang dirancang.

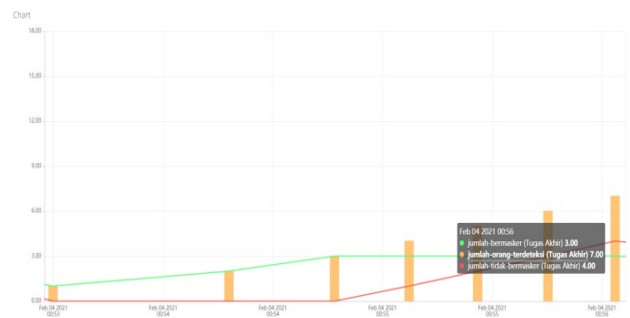
C. Pengujian Sistem Otomatis Pendeteksi Wajah Bermasker pada Raspberry Pi 4

Pengujian dilakukan pada lingkungan terkontrol dimana tempat pengujian direkayasa sedemikian rupa untuk tujuan penelitian oleh penulis sehingga dapat mendekati lingkungan saat sistem di-deploy pada lingkungan asli.

Hasil pengujian menunjukkan bahwa performa sistem dalam mendeteksi wajah bermasker lebih rendah daripada mendeteksi wajah tidak bermasker, yang dibuktikan dengan percobaan kedua dimana deteksi label *mask* terjadi kesalahan sehingga menjadi label *no mask*. Pencahayaan ruangan menjadi faktor dalam kemampuan sistem mendeteksi wajah itu sendiri yang terbukti pada percobaan ketiga dimana ada dua wajah yang tidak terdeteksi oleh sistem. Didapatkan

Tabel 9. Pengujian dengan Input Kamera pada Raspberry Pi 4.

No	Kondisi Uji	Asli			Deteksi			Error			Akurasi
		T	M	NM	T	M	NM	T	M	NM	
1	Pencahayaannya \geq 100 lux dengan kecepatan objek 0,7 m/s	7	3	4	7	2	5	0	-1	+1	85,7%
2	Pencahayaannya \geq 100 lux dengan kecepatan objek 1 m/s	10	5	5	10	3	7	0	-2	+2	80%
3	Pencahayaannya \leq 20 lux dengan kecepatan objek 0,7 m/s	7	5	2	5	2	3	-2	-3	+1	43%



Gambar 14. Visualisasi pada *Dashboard* Ubidots.

Tabel 10. Pengujian Kecepatan Komputasi Sistem.

No	Kondisi Uji	Kecepatan Komputasi Sistem (FPS)		
		TensorFlow (Normal)	TensorFlow Lite	Optimasi Kuantisasi
1.	Tidak Terjadi Deteksi	3,15	3,25	3,25
2.	Mendeteksi <i>mask</i> selama durasi kamera	1,38	2,45	2,00
3.	Mendeteksi <i>no mask</i> selama durasi kamera	1,36	2,41	2,00
4.	Tiga wajah terdeteksi selama durasi kamera	2,72	3,03	2,98

kondisi optimal lingkungan agar sistem dapat bekerja secara maksimal adalah saat pencahayaan \geq 100 lux dan kecepatan objek yang dideteksi 0,7 m/s yang mencapai akurasi sebesar 85,7%. Gambar 14 merupakan keluaran analisa sistem yang divisualisasikan dalam bentuk grafik garis dan batang pada platform IoT yang digunakan, Ubidots.

Salah satu hal yang menjadi permasalahan sistem yang menggunakan *deep learning* saat diimplementasikan pada perangkat berdaya komputasi terbatas adalah kecepatan komputasi yang dapat dilakukan oleh sistem.

Pada pengujian penelitian ini, parameter yang digunakan untuk mengukur kecepatan komputasi sistem adalah *Frame per Second (FPS)*, *frame* yang dapat diolah sistem pada satu detik. Oleh karena itu, pada subbab ini, sistem akan diuji hubungan antara parameter FPS dengan beberapa penambahan metode yaitu penggunaan pustaka *tf.lite* dan optimasi kuantisasi.

Pengukuran FPS akan dilakukan pada Raspberry Pi 4 dengan durasi pengujian masing-masing 100 detik.

Berdasarkan empat pengujian yang dilakukan, sistem yang tepat untuk dijalankan pada perangkat berdaya komputasi terbatas adalah penggunaan metode pustaka *tf.lite* pada model DL agar ukuran modelnya diperkecil menggunakan metode *flatbuffer* yang berhasil meningkatkan kecepatan proses komputasi paling tinggi sebesar 80% dari normal.

V. KESIMPULAN

Pada penelitian ini telah dibuat sistem pendeteksi otomatis wajah bermasker menggunakan *deep learning* yang diimplementasikan pada sebuah raspberry pi 4. Penggunaan sistem dibantu oleh sebuah GUI agar mudah digunakan. Hasil deteksi yang dihasilkan adalah “mask” dan “no mask” yang hasilnya berhasil dikirimkan ke Ubidots agar dilakukan visualisasi dari data yang dihasilkan. Pada pengujian, sistem mencapai akurasi 85% dari 7 objek deteksi pada kondisi optimal yaitu pencahayaan lingkungan ≥ 100 lux dan kecepatan objek deteksi 0,7 m/s serta penggunaan modul *tf.lite* yang berhasil meningkatkan kecepatan komputasi sistem paling besar 80%. Perbandingan model *deep learning* dengan metode deteksi *Haar Classifier* pada uji jenis masker, sudut deteksi, dan jarak serta pencahayaan objek menunjukkan bahwa model DL yang dibuat lebih cocok digunakan pada sistem dengan total akurasi deteksi sebesar 82%.

DAFTAR PUSTAKA

- [1] J. Howard *et al.*, “An evidence review of face masks against COVID-19,” in *Proceedings of the National Academy of Sciences*, 2021, vol. 118, no. 4, doi: 10.1073/PNAS.2014564118.
- [2] L. Tian *et al.*, “Calibrated Intervention and Containment of the COVID-19 Pandemic.” 2020, [Online]. Available: <https://arxiv.org/abs/2003.07353v6>.
- [3] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.” 2017.
- [4] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.
- [5] A. Rosebrock, “Fine-tuning with Keras and Deep Learning - PyImageSearch,” *pyimagesearch*, Jul. 2019. <https://www.pyimagesearch.com/2019/06/03/fine-tuning-with-keras-and-deep-learning/>.
- [6] W. Liu *et al.*, “SSD: Single Shot MultiBox Detector,” in *Computer Vision -- ECCV 2016*, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.
- [7] A. Oppermann, “What is Deep Learning and How does it work? | Towards Data Science.” <https://towardsdatascience.com/what-is-deep-learning-and-how-does-it-work-2ce44bb692ac> (accessed Jan. 14, 2021).
- [8] Kaggle, “Face Mask Detection.” <https://www.kaggle.com/andrewmvd/face-mask-detection> (accessed Dec. 27, 2020).
- [9] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2015, [Online]. Available: <http://www.robots.ox.ac.uk/>.
- [10] I. J. Sagina, “Why go large with Data for Deep Learning?,” Apr. 24, 2018. <https://towardsdatascience.com/why-go-large-with-data-for-deep-learning-12eee16f708> (accessed Jan. 15, 2021).
- [11] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.
- [12] D. Masters and C. Luschi, “Revisiting Small Batch Training for Deep Neural Networks.” 2018, [Online]. Available: <https://arxiv.org/abs/1804.07612v1>.
- [13] P. V. and M. Jones, “Rapid Object Detection using A Boosted Cascade of Simple Features,” 2001.